

指令 按字边界对齐存放: 指令按其数据类型大小的整数倍进行存储

e.g. int为4个字节, int只能放在首地址为0.4.8的位置上

OF 有符号数溢出标志 最高位(符号位)进位 ⊕ 次高位进位

CF 无符号数溢出标志, 计算无符号数是否超出无符号范围

浮点数运算 先把十进制数/二进制数变浮点数格式 $M \times R^E$

组相联映射

主存标记地址 $t+r$	Cache组地址 $C-r$	块内地址 b
-----------------	-------------------	-------------

Cache的组数

2^r 行(块)一组, 有 2^r 组

汇编程序填空 两个return, 有两个exit(exit1, exit2)

组相联映射

Cache目录 标记位+修改位+有效位+替换位

Cache行总位数 = 目录位 + 数据位

微程序控制器

控制信号位数: 注意控制信号不一定只有一位: 如ALUOP=0000, 有4位

第六章

1st

11. FFFFFFFF12H

13. 按字节编址,指令字长为字节整数倍

16. 32-8-4-4
操作码+通用寄存器号+基址寄存器号

操作码扩展技术

2nd

汇编语言的功能

2. 32位宿主机,数组元素占4个地址

控制器的组成	<ul style="list-style-type: none"> ① 专用寄存器: PC, IR, AR, DR ② 指令译码器 ③ 时序系统 ④ 操作控制信号形成部件
机器周期	指令周期, 机器周期, 时钟周期 { <ul style="list-style-type: none"> 1次内存读或写 1次ALU运算 1次总线传送
时钟周期	节拍, CPU主频脉冲周期, 等待节拍
CPU实现指令的方式	{ <ul style="list-style-type: none"> 单周期 多周期 流水线
微程序控制器	微程序, 微指令, 微命令, 微操作
操作控制信号 组成形成部件	{ <ul style="list-style-type: none"> ① 控制存储器 CM ② 微地址寄存器 MAR ③ 微指令寄存器 MIR ④ 后继微地址形成电路

微指令格式	控制字段：直接控制、全译码(垂直微指令格式)、字段直接编译法、字段间接编译法 判别测试字段：
微程序入口地址产生	下址字段：判定方式、计数器方式 ①映射存储器(MAPROM) ②逻辑电路
水平型微指令	控制字段采用直接控制、字段直接编译、字段间接编译
垂直型微指令	控制字段采用全译码方式
寻址方式	<div> <div>指令寻址 <div> 顺序寻址 <div> 直接转移：$TA = A$ 相对转移：$TA = PC + DISP$ 寄存器间接转移：$TA = (R_i)$ 寄存器相对转移：$TA = (R_i) + DISP$ </div> </div> </div> <div> <div>跳跃寻址</div> </div> <div> <div>数据寻址 <div> ①立即寻址 $A = imm$ ②直接寻址 $[A]$ ③间接寻址 $[[A]]$ ④寄存器寻址 $Reg[R_i]$ ⑤寄存器间接寻址 $[Reg[R_i]]$ ⑥寄存器相对寻址 <div> 变址寻址 $[Reg[R_i] + DISP]$ 基址寻址 $[Reg[R_i] + DISP]$ </div> </div> <div> ⑦相对寻址 $[PC + DISP]$ ⑧堆栈寻址 $[SP]$ </div> </div> </div> </div>

移位操作

- 逻辑移位：无符号数
- 算术移位
 - 原码：不管符号位
 - 左：数值位移出，低位补0
 - 右：低位移出，高位补0
 - 补码
 - 左：高移出，低补0
 - 右：低移出，高补符号位

偏移量
转移指令格式

下一条指令地址、符号扩展、补码

jal, jalr beq, bne, bge, blt

J I B

主机与外设交换信息的方式

- ①程序查询方式：CPU被外设独占，效率低下
- ②程序中断方式：准备好数据：中断请求
CPU执行完当前指令，中断响应，中断服务
执行IRET指令：中断返回
并行工作，效率提高
- ③直接存储器访问(DMA)，通过DMAC控制，进一步提高效率
不经过CPU(数据传递)
- ④通道与输入输出处理机方式：IOP、PPU

中断

中断源

中断向量

中断类型号

硬中断、软中断

中断服务程序的入口地址







寻找中断向量

$$(-5)_{10} = (-101)_2 = (-1.01 \times 2^{+10})_2$$

$$M_s = 1, E = +10 \quad M = 10100000 = 01000000$$

$$= (10000001)_{\text{隐藏}} \text{ 移}$$

$$(-5)_{\text{BF16}} = 1100000010100000$$

	有隐藏位,化成隐藏位格式
逻辑门	与  与非  或  或非  非  异或 
中断请求	~屏蔽 ~INTM 外中断才有, 中断请求触发器INTR, 中断请求寄存器IRR, 中断字 解决的问题: 中断屏蔽、中断请求信号的传递、CPU对中断请求信号的监测
中断响应	执行中断隐指令, 硬件保存程序断点及标志寄存器 解决: 中断优先级判别、中断源识别 中断排队、判优电路 向量方式/软件查询方式
中断服务	保护现场、处理中断、恢复现场 解决: 中断嵌套(多重中断)
中断返回	执行IRET, 回到CPU程序断点处
中断响应过程	中断优先级 硬 > 软 > I/O 排队判优: 软件查询、硬件排队电路 关中断、保存断点(PC、PSW)、识别中断源

$$E_x = 1101 = +101 = 5$$

$$E_y = +010 = 2$$

$$E_y < E_x$$

$$Y \text{ 对阶 } X, M_y = 1.0011101011 \quad M_x = 0.1110100000$$

$$= 1.1110011101(11)$$

$$[M_x - M_y]_{\text{补}} = [M]_{\text{补}} + [-M_y]_{\text{补}}$$

$$[-M_y]_{\text{补}} = 0.0001100101(001)$$

$$\begin{array}{r} 0.0.1111010000 \\ + 0.0001100101 \\ \hline 0.1.0001100101(001) \end{array}$$

$$\text{溢出, 右规 } \times 1, 0.1000110101(001) \quad 5+1=6$$

$$= 0.1000110101 \times 2^6$$

$$0.1000110101 \neq 1110$$

浮点数加减
运算

① 化成规格化数

② 对阶, 保留舍入位)

③ 结果规格化

溢出: 右规 $\times 1$, 阶码 $+1$

有前导零: 左规, 阶码 $-n$

④ 舍入